

# Python Fun(damentals)

## Python Text Sequences

Alexander Rymdeko-Harvey

Obscurity Labs

\* Text Sequence (Strings)



# Python Text Sequence Operator Cont.

## String `+` Operator

```
>>> a = 'alex' + 'rymdekeo-harvey'  
>>> b = 'alex' + ' rymdeko-harvey'  
>>> print(a)  
alexrymdekeo-harvey  
>>> print(b)  
alex rymdeko-harvey
```

## String `in` Operator

```
>>> 'alex' in 'alex rymdeko-harvey'  
True
```

# Python Text Sequence Operator Cont.

## String Indexing

```
>>> a = 'ALEX'  
>>> print(a[1])  
L
```

What does this look like in C/C++?

```
// valid initialization  
char name[10] = {'A', 'L', 'E', 'X', '\0'};
```

What does this look like in the Python Object?

A	L	E	X
---	---	---	---

# Python Text Sequence Methods

- Strings implement all of the common sequence operations
- With the additional methods
- Python provides TONS of string methods we will only cover a few but labs may require research
- The most common style is Format String Syntax

# Python Text Sequence Methods Cont.

Here is a example of `str.capitalize()` :

```
# Return a copy of the string with its first
# character capitalized and the rest lowercased.
>>> a = 'alex'
>>> a.capitalize()
'Alex'
```

Here is a example of `str.capitalize()` :

```
# Return a copy of the string with its first
# character capitalized and eading characters removed.
>>> a = '  alex  '
>>> a.lstrip().capitalize()
'Alex  '
```

<https://docs.python.org/3/library/stdtypes.html#str.capitalize>

<https://docs.python.org/3/library/stdtypes.html#str.lstrip>

# Lab\_2.py

## TASKING

Perform the following on the variable `dataStr` :

1. Set `dataStr` value to `'opensource.com'`
2. Set `dataStrFull` full value of `https://` and use the `dataStr` to create this variable
3. Set `dataStrSplit` to the value of `dataStrFull` split on the character `.`, into a list using a string method (*HINT: Python Docs*)
4. Try to print `dataStrSplit` with the format string methods like `print('{} {}'.format('one', 'two'))`